

**Amendments to the Claims:**

This listing of claims replaces all prior versions and listings of claims in the application.

**Listing of Claims:**

1. (Currently Amended) A method for handling transaction messages in asynchronous data replication in a database system, the database system including a source node and a target node, each transaction message having information concerning a row change to a table copy at the source node, the method comprising:

determining whether a first transaction message has a dependency on a preceding non-completed transaction message, the first transaction message having a dependency on the preceding non-completed transaction when a row change associated with the preceding non-completed transaction requires application to a table copy at the target node prior to a row change associated with the first transaction message;

responsive to the first transaction message having a dependency on the preceding non-completed transaction,

holding the first transaction message;

completing the preceding non-completed transaction message including applying the row change associated with the preceding non-completed transaction message to the table copy at the target node; and

responsive to completing the preceding non-completed transaction message, releasing the first transaction message and applying the row change associated with the first transaction message to the table copy at the target node; and

responsive to the first transaction message not having a dependency on the preceding non-completed transaction, applying the row change associated with the first transaction message to the table copy at the target node without holding the first transaction message,

wherein determining whether the first transaction message has a dependency on the preceding non-completed transaction message comprises

determining that the row change in the first transaction message is an insert or a key update type of change;

comparing a new replication key value in the row change in the first transaction message to an old replication key value of a row change in the preceding non-completed transaction message, including comparing a hash value of the new replication key value in the row change in the first transaction message to a hash value of the old replication key value in the row change in the preceding non-completed transaction message; and

determining that the first transaction message has a dependency on the preceding non-completed transaction message if the new replication key value in the row change in the first transaction message is the same as the old replication key value in the row change in the preceding non-completed transaction message.

2. (Previously Presented) The method of claim 1, further comprising:

examining a plurality of transaction messages on a work queue by a plurality of agent threads;

applying in parallel row changes in each of the plurality of transaction messages by each of the plurality of agent threads;

updating a control table to indicate completion of the application of each of the plurality of transaction messages; and

placing each completed transaction message on a done queue.

3. (Previously Presented) The method of claim 2, further comprising:

examining each completed transaction message on the done queue;

determining if completion of a completed transaction message clears dependencies of any held transaction messages dependent upon the completed transaction message; and

placing any of the held transaction messages onto the work queue, if the dependencies of the held transaction message have been cleared.

4-5. (Cancelled)

6. (Previously Presented) The method of claim 1, wherein determining whether the first transaction message has a dependency on the preceding non-completed transaction message comprises:

determining that the row change in the first transaction message is a delete or a key update type of change;

comparing an old replication key value in the row change in the first transaction message to a new replication key value in a row change in the preceding non-completed transaction message; and

determining that the first transaction message has a dependency on the preceding non-completed transaction message if the old replication key value in the row change in the first

transaction message is the same as the new replication key value in the row change in the preceding non-completed transaction message.

7. (Previously Presented) The method of claim 6, wherein comparing an old replication key value in the row change in the first transaction message to a new replication key value in a row change in the preceding non-completed transaction message comprises:

comparing a hash value of the old replication key value in the row change in the first transaction message to a hash value of the new replication key value in the row change in the preceding non-completed transaction message.

8. (Previously Presented) The method of claim 1, wherein determining whether the first transaction message has a dependency on the preceding non-completed transaction message comprises:

determining that the row change in the first transaction message is an update type of change;

comparing a new replication key value in the row change in the first transaction message to a new replication key value in a row change in the preceding non-completed transaction message; and

determining that the first transaction message has a dependency on the preceding non-completed transaction message if the new replication key value in the row change in the first transaction message is the same as the new replication key value in the row change in the preceding non-completed transaction message.

9. (Previously Presented) The method of claim 8, wherein comparing a new replication key value in the row change in the first transaction message to a new replication key value in a row change in the preceding non-completed transaction message comprises:

comparing a hash value of the new replication key value in the row change in the first transaction message to a hash value of the new replication key value in the row change in the preceding non-completed transaction message.

10. (Previously Presented) The method of claim 2, further comprising:

removing the completed transaction message from a receive queue.

11. (Previously Presented) The method of claim 10, wherein removing the completed transaction message from the receive queue comprises:

deleting the completed transaction message from the receive queue as part of a two-phase commit synchronization with the application of the completed transaction message.

12. (Previously Presented) The method of claim 10, wherein removing the completed transaction message from the receive queue comprises:

obtaining at least one entry in a control table at the target node indicating that the completed transaction message has been completed; and

deleting the completed transaction message from the receive queue.

13. (Previously Presented) The method of claim 12, further comprising:

removing the at least one entry from the control table.

14. (Currently Amended) A computer readable medium containing a computer program for handling transaction messages in asynchronous data replication in a database system, the database system including a source node and a target node, each transaction message having information concerning at least one row change to a table copy at the source node, the computer program comprising programming instructions for:

determining whether a first transaction message has a dependency on a preceding non-completed transaction message, the first transaction message having a dependency on the preceding non-completed transaction when a row change associated with the preceding non-completed transaction requires application to a table copy at the target node prior to a row change associated with the first transaction message;

responsive to the first transaction message having a dependency on the preceding non-completed transaction,

holding the first transaction message;

completing the preceding non-completed transaction message including applying the row change associated with the preceding non-completed transaction message to the table copy at the target node; and

responsive to completing the preceding non-completed transaction message, releasing the first transaction message and applying the row change associated with the first transaction message to the table copy at the target node; and  
responsive to the first transaction message not having a dependency on the preceding non-completed transaction, applying the row change associated with the first transaction message to the table copy at the target node without holding the first transaction message;

wherein the programming instructions for determining whether the first transaction message has a dependency on the preceding non-completed transaction message comprises programming instructions for:

determining that the row change in the first transaction message is an insert or a key update type of change;

comparing a new replication key value in the row change in the first transaction message to an old replication key value of a row change in the preceding non-completed transaction message, including comparing a hash value of the new replication key value in the row change in the first transaction message to a hash value of the old replication key value in the row change in the preceding non-completed transaction message; and

determining that the first transaction message has a dependency on the preceding non-completed transaction message if the new replication key value in the row change in the first transaction message is the same as the old replication key value in the row change in the preceding non-completed transaction message.

15. (Previously Presented) The computer readable medium of claim 14, wherein the computer program further comprises programming instructions for:

examining a plurality of transaction messages on a work queue by a plurality of agent threads;

applying in parallel row changes in each of the plurality of transaction messages by each of the plurality of agent threads;

updating a control table to indicate completion of the application of each of the plurality of transaction messages; and

placing each completed transaction message on a done queue.

16. (Previously Presented) The computer readable medium of claim 15, wherein the computer program further comprises programming instructions for:

examining each completed transaction message on the done queue;

determining if completion of a completed transaction message clears dependencies of any held transaction messages dependent upon the completed transaction message; and

placing any of the held transaction messages onto the work queue, if the dependencies of the held transaction message have been cleared.

17-18. (Cancelled)

19. (Previously Presented) The computer readable medium of claim 14, wherein the programming instructions for determining whether the first transaction message has a dependency on the preceding non-completed transaction message comprises programming instructions for:

determining that the row change in the first transaction message is a delete or a key update type of change;

comparing an old replication key value in the row change in the first transaction message to a new replication key value in a row change in the preceding non-completed transaction message; and

determining that the first transaction message has a dependency on the preceding non-completed transaction message if the old replication key value in the row change in the first



transaction message is the same as the new replication key value in the row change in the preceding non-completed transaction message.

20. (Previously Presented) The computer readable medium of claim 19, wherein the programming instructions for comparing an old replication key value in the row change in the first transaction message to a new replication key value in a row change in the preceding non-completed transaction message comprises programming instructions for:

comparing a hash value of the old replication key value in the row change in the first transaction message to a hash value of the new replication key value in the row change in the preceding non-completed transaction message.

21. (Previously Presented) The computer readable medium of claim 14, wherein the programming instructions for determining whether the first transaction message has a dependency on the preceding non-completed transaction message comprises programming instructions for:

determining that the row change in the first transaction message is an update type of change;

comparing a new replication key value in the row change in the first transaction message to a new replication key value in a row change in the preceding non-completed transaction message; and

determining that the first transaction message has a dependency on the preceding non-completed transaction message if the new replication key value in the row change in the first

transaction message is the same as the new replication key value in the row change in the preceding non-completed transaction message.

22. (Currently Amended) The computer readable medium of claim 21, wherein the programming instructions for comparing ~~comparing~~ a new replication key value in the row change in the first transaction message to a new replication key value in a row change in the preceding non-completed transaction message comprises programming instructions for:

comparing a hash value of the new replication key value in the row change in the first transaction message to a hash value of the new replication key value in the row change in the preceding non-completed transaction message.

23. (Previously Presented) The computer readable medium of claim 15, wherein the computer program further comprises programming instructions for:

removing the completed transaction message from a receive queue.

24. (Previously Presented) The computer readable medium of claim 23, wherein the programming instructions for removing the completed transaction message from the receive queue comprises programming instructions for:

deleting the completed transaction message from the receive queue as part of a two-phase commit synchronization with the application of the completed transaction message.

25. (Previously Presented) The computer readable medium of claim 23, wherein the programming instructions for removing the completed transaction message from the receive queue comprises programming instructions for:

obtaining at least one entry in a control table at the target node indicating that the completed transaction message has been completed; and

deleting the completed transaction message from the receive queue.

26. (Previously Presented) The computer readable medium of claim 25, wherein the computer program further comprises programming instructions for:

removing the at least one entry from the control table.

27-33. (Cancelled)